STAT 672 Statistical Learning & Data Analytics Spring 2018

George Mason University

Instructor: Martin Slawski

Homework 3

Pencil-and-paper part due Thursday, March 22nd, in class. Code and relevant output has to be submitted via Blackboard. Please follow the following format for the filename of your submission: `Lastname_Firstname_HWx.zip`, where `x` needs to be substituted by the homework #.
Maximum possible points to earn: 20; 100% = 15 points, rest counts as bonus.

Ridge Regression vs. LASSO

The goal of this problem is to compare the performance of ridge regression and the LASSO for predicting the popularities of topics in a social network. The target variable is "#active discussions" given 77 predictor variables such as number of authors contributing to the topic over time, average discussion lengths, number of interactions between authors etc. Several of the original predictor variables as well as the target variable have been log-transformed prior to analysis.

a) Download the files `hw3_X.csv` and `hw_3Y.csv` from Blackboard. The first file contains the data $\mathbf{X}$ for the predictors, the second file the the data for the target variable $\mathbf{y}$. We have $n = 8,000$.

b) We are going to work with a quadratic model: form a matrix $\mathbf{\Phi}$ from $\mathbf{X}$ by including all quadratic terms and all first-order interactions. The matrix $\mathbf{\Phi}$ has dimensions $n = 8,000$ and $D = 3,080$.

c) Split $(\mathbf{\Phi}, \mathbf{y})$ into a training set $(\mathbf{\Phi}_{\text{train}}, \mathbf{y}_{\text{train}})$ and a test set $(\mathbf{\Phi}_{\text{test}}, \mathbf{y}_{\text{test}})$ by assigining observations $4, 8, 12, \ldots, 8,000$ to the test set.

d) Center and scale the training set:

1) Centering.

$$y_{\text{train},i} \leftarrow y_{\text{train},i} - \overline{y}_{\text{train}}, \;\; i \in \text{train}, \qquad \text{where} \;\; \overline{y}_{\text{train}} := \sum_{i' \in \text{train}} y_{i'} \Big/ n_{\text{train}},$$

$$\mathbf{\Phi}_{i,j} \leftarrow \mathbf{\Phi}_{i,j} - \overline{\mathbf{\Phi}}_j, \qquad i \in \text{train}, \qquad \text{where} \;\; \overline{\mathbf{\Phi}}_j := \sum_{i' \in \text{train}} \mathbf{\Phi}_{i',j} \Big/ n_{\text{train}}, \;\; j = 1, \ldots, D.$$

*2) Scaling (*after* Centering).*

$$\mathbf{\Phi}_{i,j} \leftarrow \frac{\mathbf{\Phi}_{i,j}}{\zeta_j}, \quad i \in \text{train}, \qquad \zeta_j := \sqrt{\sum_{i' \in \text{train}} \mathbf{\Phi}_{i',j}^2}, \;\; j = 1, \ldots, D,$$

Hint. In Python/R It is possible do this without a double loop.

After the pre-processing in step d), we want to fit ridge regression:

$$\|\mathbf{y}_{\text{train}} - \mathbf{\Phi}_{\text{train}} w\|_2^2 + \lambda \|w\|_2^2, \tag{1}$$

for $\lambda \in \Lambda_{\text{ridge}} = \{2^{-13}, 2^{-12.5}, 2^{-12}, \ldots, 2^{8.5}, 2^9\}$.

e) Computationally efficient ridge regression for multiple choices of λ:

Naively looping through all values in Λ_{ridge} is highly inefficient. Instead we are going to do the following:

1. Compute the singular value decomposition (SVD) of $\mathbf{\Phi}_{\text{train}}$:

   $$\mathbf{\Phi}_{\text{train}} = \mathbf{U}\mathbf{S}\mathbf{V}^\top,$$

   where $\mathbf{U} \in \mathbb{R}^{n_{\text{train}} \times D}$, $\mathbf{U}^\top\mathbf{U} = I_D$, $\mathbf{S} \in \mathbb{R}^{D \times D}$ is diagonal, and $\mathbf{V} \in \mathbb{R}^{D \times D}$, $\mathbf{V}\mathbf{V}^\top = \mathbf{V}^\top\mathbf{V} = I_D$.

   Hint. Use `svd` in R or `numpy.linalg.svd` in Python. Note that for the latter the SVD is parameterized slightly differently ($\mathbf{V}$ corresponds to $\mathbf{V}^\top$).

2. Denote by $\widehat{w}_{\text{ridge}}(\lambda)$ the minimizer of (1). Show the **second** equality in

   $$\begin{aligned}\widehat{w}_{\text{ridge}}(\lambda) &= (\mathbf{\Phi}_{\text{train}}^\top \mathbf{\Phi}_{\text{train}} + \lambda I_D)^{-1} \mathbf{\Phi}_{\text{train}}^\top \mathbf{y}_{\text{train}} \\ &= \mathbf{V}\mathbf{S}_\lambda \mathbf{U}^\top \mathbf{y}_{\text{train}}, \quad \mathbf{S}_\lambda := \text{diag}\left(\frac{s_1}{s_1^2 + \lambda}, \ldots, \frac{s_D}{s_D^2 + \lambda}\right),\end{aligned}$$

   where s_j, $j = 1, \ldots, D$, are the diagonal entries of $\mathbf{S}$.

3. Deduce that once the SVD has been obtained, computing $\widehat{w}_{\text{ridge}}$ can be done in $O(D^2)$ flops. For a new value of λ we only need to update the matrix $\mathbf{S}_\lambda$, but do not have to solve a new linear system (which would require $O(nD^2)$ flops).

f) Compute the test error of ridge regression as λ varies:

$$\frac{1}{n_{\text{test}}} \|\mathbf{y}_{\text{test}} - \mathbf{1}_{n_{\text{test}}} \widetilde{w}_{0,\text{ridge}}(\lambda) - \mathbf{\Phi}_{\text{test}} \widetilde{w}_{\text{ridge}}(\lambda)\|_2^2,$$

where $\mathbf{1}_{\text{ntest}}$ is a vector of n_{test} ones, and

$$\widetilde{w}_{\text{ridge},j}(\lambda) = \widehat{w}_{\text{ridge},j}/\zeta_j, \quad j = 1, \dots, D, \qquad \widetilde{w}_{0,\text{ridge}}(\lambda) = \overline{y}_{\text{train}} - \sum_{j=1}^{D} \widetilde{w}_{\text{ridge},j}(\lambda)\overline{\mathbf{\Phi}}_j. \quad (2)$$

The introduction of the intercept $\widetilde{w}_{0,\text{ridge}}$ and the use of the re-scaled ridge coefficients $\widehat{w}_{\text{ridge}}$ are used to account for centering and scaling performed in d).

g) Next, we try the LASSO instead of ridge regression.

$$\min_w \frac{1}{2n_{\text{train}}} \|\mathbf{y}_{\text{train}} - \mathbf{\Phi}_{\text{train}} w\|_2^2 + \lambda \|w\|_1 \quad (3)$$

Note the constant $\frac{1}{2n_{\text{train}}}$ in front of the fitting term.

Try $\lambda \in \Lambda_{\text{lasso}} = \sqrt{\frac{\log D}{n_{\text{train}}}} \cdot \{2^{-13}, 2^{12}, \dots, 2^0, 2^1\}$.

Recommended software for fitting:
R: glmnet, Python: `linear_model.Lasso` in scikit-learn. It is recommended to specify Λ as a *decreasing sequence* and to use warm starts.

Carefully check and set the input parameters. In particular, find out the exact form of the lasso objective that is minimized. Depending on that, you may have to re-scale the regularization parameter accordingly. For example, if the software uses

$$\min_w \frac{1}{n_{\text{train}}} \|\mathbf{y}_{\text{train}} - \mathbf{\Phi}_{\text{train}} w\|_2^2 + \lambda \|w\|_1$$

you have to work with $2 \cdot \Lambda_{\text{lasso}}$ instead of Λ_{lasso}.

Depending on your machine, you may have to wait for a few minutes to get the results.

h) Analagously to f), compute the test errors as λ varies:

$$\frac{1}{n_{\text{test}}} \|\mathbf{y}_{\text{test}} - \mathbf{1}_{n_{\text{test}}} \widetilde{w}_{0,\text{lasso}}(\lambda) - \mathbf{\Phi}_{\text{test}} \widetilde{w}_{\text{lasso}}(\lambda)\|_2^2,$$

where in analogy to (2)

$$\widetilde{w}_{\text{lasso},j}(\lambda) = \widehat{w}_{\text{lasso},j}/\zeta_j, \quad j = 1, \dots, D, \qquad \widetilde{w}_{0,\text{lasso}}(\lambda) = \overline{y}_{\text{train}} - \sum_{j=1}^{D} \widetilde{w}_{\text{lasso},j}(\lambda)\overline{\mathbf{\Phi}}_j,$$

with $\widehat{w}_{\text{lasso}}(\lambda)$ denoting the minimizer of (3).

i) Plot the test errors of f) and h). Which method achieves the better minimum test error, where the minimum is taken over λ?

Point distribution:

a)	b)	c)	d)	e)	f)	g)	h)	i)
1	1	-	2	1 + 4 + 2	4	-	3	2